[illegible]

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/334,104	06/16/1999	GALEN C. HUNT	MS1-354US	4938

22801 7590 03/11/2004

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER
ANYA, CHARLES E

ART UNIT	PAPER NUMBER
2126	18

DATE MAILED: 03/11/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

PRg

Office Action Summary	Application No. 09/334,104	Applicant(s) HUNT ET AL.	
	Examiner Charles E Anya	Art Unit 2126	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *11 December 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-41* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-41* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-41 are pending in this application.

2. Please furnish the office with the references (Windows 95 Win32 Programming API Bible by Richard Simon and Inside OLE by Kraig Brockschmidt) cited on pages 6 and 11 of the instant application.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. **Claims 1-3,6-17,19,22,23,29-31 and 36-41 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,379,432 to Orton et al. in view of Understanding ActiveX and OLE to David Chappell, pages 4-5.**

5. As to claim 1, Orton teaches a method of factoring operating system functions comprising: defining criteria that governs how functions of an operating system are to be factored into one or more groups factoring the functions into one or more groups based upon the criteria (figure 4 Col. 7 Ln. 1 – 67, Col. 8 Ln. 1 – 14) and associating groups of functions with programming objects that have data and methods, wherein the methods

correspond to the operating system functions effective to provide an object oriented operating system ("...method..." Col. 7 Ln. 1 – 12).

Although Orton teaches library server (figure 3) it is not explicit with reference to the programming objects being configured to be instantiated throughout a remote computing system.

Chappell teaches the programming objects being configured to be instantiated throughout a remote computing system ("...network..." pages 4 –5, pages 245-246). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification to provide one application's services to another (page 4 lines 22 – 35).

6. As to claim 2, Orton teaches the method of claim 1, wherein the programming objects have interfaces through which the methods can be accessed ("...method..." Col. 7 Ln. 1 – 12).

7. As to claim 3, Orton is silent with reference to the method of claim 1, wherein the programming objects comprise COM objects.

Chappell teaches the method of claim 1, wherein the programming objects comprise COM objects ("COM object..." page 8). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification in order to use the object's services (page 8 lines 29 – 30).

8. As to claim 6, Orton teaches the method of claim l further comprising instantiating a plurality of programming objects across a process boundary ("...task address space..." Col. 9 Ln. 12 – 31).

9. As to claim 7, Orton is silent with reference to the method of claim 1, further comprising instantiating a plurality of programming objects across a machine boundary. Chappell teaches the method of claim 1, further comprising instantiating a plurality of programming objects across a machine boundary ("...network..." pages 4 –5, pages 245-246). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification to provide one application's services to another (page 4 lines 22 – 35).

10. As to claim 8, Orton teaches the method of claim 1, wherein the criteria are based, at least in part, on the manner in which particular functions behave (Col. 7 Ln. 30 – 67, Col. 1 – 14).

11. As to claim 9, Orton teaches the method of claim 8, wherein the manner includes a consideration of the types of operating System resources that are associated with the operation of a function (Col. 7 Ln. 30 – 67, Col. 1 – 14).

12. As to claim 10, Orton teaches the method of claim 8, wherein the manner includes a consideration of whether a particular function creates an operating resource (Col. 7 Ln. 39 – 42).

13. As to claim 11, Orton teaches the method of claim 8, wherein the manner includes a consideration of whether a particular function operates upon an operating system resource (Col. 7 Ln. 30 – 67, Col. 1 – 14).

14. As to claim 12, Orton teaches the method of claim 1, wherein the criteria is based, at least in part, on the manner in which particular functions behave, wherein the manner includes: a consideration of the types of operating system resources that are, associated with the operation of a function; and to a consideration of whether a particular function creates an operating system resource (Col. 7 Ln. 30 – 67, Col. 1 – 14).

15. As to claim 13, Orton teaches the method of claim 1, wherein the criteria is based, at least in part, on the manner in which particular functions behave, wherein the manner includes: a consideration of the types of operating system resources that are associated with the operation of a function call; and a consideration of whether a particular function operates upon a given operating system resource (Col. 7 Ln. 30 – 67, Col. 1 – 14).

16. As to claim 14, see the rejection of claim 1.

17. As to claim 15, Orton teaches the method of claim 14, wherein the first criteria are based upon the type of resource that is associated with an operation of a function (Col. 7 Ln. 30 – 67, Col. 1 – 14).

18. As to claim 16, Orton teaches the method of claim 14, wherein the second criteria are based upon the nature of an operation of a function on a particular resource (Col. 7 Ln. 30 – 67, Col. 1 – 14).

19. As to claim 17, Orton teaches the method of claim 16, wherein said nature concerns whether a function creates a resource (Col. 7 Ln. 39 – 42).

20. As to claim 19, Orton teaches the method of claim 14, wherein the first criteria is based upon the type of resource that is associated with an operation of a function, and the particular resource (Col. 7 Ln. 30 – 67, Col. 1 – 14).

21. As to claim 22, see the rejection of claim 6.

22. As to claim 23, see the rejection of claim 7.

23. As to claim 29, Orton teaches an operating system application program interface embodied on a computer-readable medium comprising a plurality of object interfaces is associated with an object that object interfaces (figure 4 Col. 7 Ln. 39 – 67, Col. 8 Ln. 1 – 14), wherein each object interface includes one or more methods that are associated and can call functions of an operating system that does not comprise the object interfaces (figure 2/3 Col. 8 Ln. 15 – 67, Col. 9 Ln. 1 – 67, Col. 10 Ln. 1 – 62).
Although Orton teaches a library server (figure 3) it is not explicit with reference to at least one the objects being configured to be remotely instantiated.
Chappell teaches at least one the objects being configured to be remotely instantiated ("...network..." pages 4 –5, pages 245-246). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification to provide one application's services to another (page 4 lines 22 – 35).

24. As to claim 30, see the rejection of claim 9.

25. As to claim 31, see the rejection of claim 10.

26. As to claim 36, Orton teaches an operating system comprising: a plurality of programming objects having interfaces, wherein the programming objects represent operating system resources (figure 4 Col. 7 Ln. 39 – 67, Col. 8 Ln. 1 – 14) and wherein the interfaces define methods that are organized in accordance with whether they

create an operating system resource or not wherein the programming objects are configured to be called either directly or indirectly by an application; and wherein the methods are configured to call operating system functions responsive to being called directly or indirectly by an application (figure 2/3 Col. 8 Ln. 15 – 67, Col. 9 Ln. 1 – 67, Col. 10 Ln. 1 – 62).

Although Orton teaches a library server (figure 3) it is silent with reference to the programming objects being configured to be instantiated throughout a remote computing system.

Chappell teaches the programming objects being configured to be instantiated throughout a remote computing system ("…network…" pages 4 –5, pages 245-246). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification to provide one application's services to another (page 4 lines 22 – 35).

27. As to claim 37, see the rejection of claim 6.

28. As to claim 38, see the rejection of claim 7.

29. As to claim 39, see the rejection of claims 6 and 7.

30. As to claim 40, see the rejection of claim 3.

31. As to claim 41, Orton teaches a method of converting an operating system from a non-object-oriented format to an object oriented format, wherein the operating system includes a plurality of operating system functions that are callable to create or use operating system resources, the method comprising defining a plurality of programming object interfaces that define methods that correspond to the operating system functions, wherein programming objects that support the interfaces are callable either directly by an application that makes object-oriented calls, or indirectly by an application that makes function calls (figure 4 Col. 7 Ln. 1 – 67, Col. 1 – 14), calling a programming object interface either directly via an object-oriented call, or indirectly via an indirection that transforms a function call into an object-oriented call; and responsive to said calling, calling an operating system function with a method of the programming object that supports said programming object interface (figure 2/3 Col. 8 Ln. 15 – 67, Col. 9 Ln. 1 – 67, Col. 10 Ln. 1 – 62).

Although Orton teaches a library server (figure 3) it is not explicit with reference to the programming objects being configured to be instantiated throughout a remote computing system.

Chappell teaches the programming objects being configured to be instantiated throughout a remote computing system ("...network..." page 4 –5, pages 245-246). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification to provide one application's services to another (page 4 lines 22 – 35).

32. Claims 4,5,18,20,21,24-28 and 32-35 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Pat. No. 5,379,432 to Orton et al. in view of Understanding ActiveX and OLE to David Chappell, pages 4-5 as applied to claim 1 above, and further in view of U.S. Pat. 6,33,157 B1 to Oppermann et al.

33. As to claim 4, Orton as modified is silent with reference to the method of claim 1, wherein the factoring comprises creating a hierarchy of object interfaces in which certain interfaces can inherit from other interfaces.

Oppermann teaches the method of claim 1, wherein the factoring comprises creating a hierarchy of object interfaces in which certain interfaces can inherit from other interfaces (figure 3 Col. 8 Ln. 16 – 67). It would have been obvious to apply the teaching of Oppermann to the system of Orton. One would have been motivated to make such modification in order to save memory (Col. 8 Ln. 60 – 67).

34. As to claim 5, Orton as modified is silent with reference to the method of claim 1, wherein said factoring comprises creating a hierarchy of object interfaces in which certain interfaces can aggregate with other interfaces.

Oppermann teaches the method of claim 1, wherein said factoring comprises creating a hierarchy of object interfaces in which certain interfaces can aggregate with other interfaces (IUKnown Col. 9 Ln. 1 – 22). It would have been obvious to apply the teaching of Oppermann to the system of Orton. One would have been motivated to make such modification to provide interface location transparency.

35. As to claim 18, Orton as modified is silent the method of claim 16, wherein said nature concerns whether a function does not create a resource.
Oppermann teaches the method of claim 16, wherein said nature concerns whether a function does not create a resource ("...default action..." Col. 8 Ln. 38 – 59). It would have been obvious to apply the teaching of Oppermann to the system of Orton. One would have been motivated to make such modification to provide button function.

36. As to claim 20, see the rejection of claim 4.

37. As to claim 21, see the rejection of claim 5.

38. As to claim 24, Orton teaches a method of factoring operating system functions comprising: factoring a plurality of operating system functions into interface groups based upon the resources with which a function is associated (figure 4 Col. 7 Ln. Ln. 1 – 67, Col. 8 Ln. 1 – 14).
Although Orton teaches library server (figure 3) it is not explicit with reference to individual interface sub-groups being associated with individual programming objects that can be instantiated throughout a remote computing system and factoring the groups into interface sub-groups based upon each function's handle that represents a resource and organizing the interface sub-groups so that at least one of the interface sub-groups inherits from at least one other of the interface sub-groups.

Chappell teaches individual interface sub-groups being associated with individual programming objects that can be instantiated throughout a remote computing system ("...network..." pages 4 – 5, pages 245 – 246). It would have been obvious to apply the teaching of Chappell to the system of Orton. One would have been motivated to make such a modification to provide one application's services to another (page 4 lines 22 – 35).

Oppermann teaches factoring the groups into interface sub-groups based upon each function's handle that represents a resource and organizing the interface sub-groups so that at least one of the interface sub-groups inherits from at least one other of the interface sub-groups (figure 3 Col. 8 Ln. 16 – 67, "...LResult..." Col. 21 Ln. 35 – 51). It would have been obvious to apply the teaching of Oppermann to the system of Orton. One would have been motivated to make such modification in order to save memory (Col. 8 Ln. 60 – 67).

39. As to claim 25, see the rejection of claim 5

40. As to claim 26, see the rejection of claim 3.

41. As to claim 27, Orton as modified is silent with reference to the method of claim 24, wherein the factoring of the interface groups into interface sub-groups comprises considering whether a function creates a handle.

Oppermann teaches the method of claim 24, wherein the factoring of the interface groups into interface sub-groups comprises considering whether a function creates a handle (Col. 21 Ln. 30 – 54). It would have been obvious to apply the teaching of Oppermann to the system of Orton. One would have been motivated to make such modification in order to receive a pointer (Col. 21 Ln. 40 – 45).

42. As to claim 28, Orton as modified is silent with reference to the method of claim 24, wherein said organizing comprises aggregating at least one of the interface sub-groups, wherein the factoring of the interface groups into interface sub-groups comprises considering whether a function call creates a handle.
Oppermann teaches the method of claim 24, wherein said organizing comprises aggregating at least one of the interface sub-groups, wherein the factoring of the interface groups into interface sub-groups comprises considering whether a function call creates a handle (Col. 22 Ln. 15 – 36). It would have been obvious to apply the teaching of Oppermann to the system of Orton. One would have been motivated to make such modification to provide screen location.

43. As to claim 32, see the rejection of claim 18.

44. As to claim 33, see the rejection of claim 18.

45. As to claim 34, see the rejection of claim 4.

46. As to claim 35, see the rejection of claim 5.

Response to Arguments

47. Applicant's arguments with respect to claims 1-41 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

48. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. Pat. No. 4,768,150 to Chang et al.

U.S. Pat. No. 6,081,807 to Story et al.

U.S. Pat. No. 6,317,773 B1 to Cobb et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Charles E Anya whose telephone number is (703) 305-3411. The examiner can normally be reached on M-F (8:30-6:00) First Friday off.

The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles E Anya
Examiner
Art Unit 2126

cea

[illegible]
MENG-AI T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100